**🌆 Sustainable Smart City Assistant – Project Workflow**

**💡 Overview:**

An AI-powered smart city dashboard using **IBM Watsonx Granite LLM**, **FastAPI backend**, and **Streamlit frontend**, enabling citizens and city officials to monitor sustainability KPIs, summarize policies, ask eco-related questions, and interact with AI-generated insights.

**🚦 1. User Input & Interactions (Streamlit Frontend)**

Users can interact via a responsive and visually enhanced Streamlit interface to:

* 🧠 **Submit prompts** for sustainability Q&A or policy summaries.
* 📄 **Upload documents** (.txt, .csv) for summarization and vector search.
* 🌍 **Choose a city** to monitor real-time KPIs (e.g., water usage, air quality).
* 📝 **Submit feedback** with name, category, and message.
* 💬 **Ask sustainability queries** in a chat interface.
* 🌱 **Search eco tips** using keywords.

**🔧 UI Components:**

* smart\_dashboard.py
* feedback\_form.py
* chat\_assistant.py
* eco\_tips.py
* summary\_card.py

**⚙️ 2. Backend Processing (FastAPI)**

Each frontend interaction sends API requests to specific FastAPI endpoints:

* 📥 **Feedback Handling** → feedback\_router.py
* 📊 **KPI Forecasting** from .csv → kpi\_file\_forecaster.py
* 🤖 **Text Processing (LLM)** → granite\_llm.py
* 📈 **Anomaly Detection** for uploaded datasets → Statistical checks
* 🔎 **Policy Document Search** → vector\_router.py using Pinecone

**📁 Backend Modules:**

* vector\_router.py
* chat\_router.py
* kpi\_upload\_router.py
* granite\_llm.py
* pinecone\_client.py

**🧠 3. AI & ML Processing**

* ✨ **Watsonx Granite LLM**: Generates responses to queries, summaries, and eco tips in natural language.
* 🔬 **Forecasting Models**: Predict future city KPIs.
* 🚨 **Anomaly Detection**: Flags irregularities in uploaded KPI datasets.
* 🧠 **Vector Search**: Semantic similarity search over document chunks via **Pinecone**.

**📦 Output Format:**

* JSON objects containing:
  + Chat response
  + KPI forecast
  + Policy summary
  + Anomaly alert
  + Eco tips

**🖥️ 4. Frontend Display (Streamlit)**

Rendered features in real-time:

* 📈 **Summary Cards**: City KPIs visualized using summary\_card.py
* 🤖 **AI Responses**: Displayed directly under chat, tips, or summary inputs
* 📄 **Search Results**: User-readable policy document chunks
* ✅ **User Feedback**: Toast notifications for success/failure states

**🎨 Frontend Enhancements:**

* Rounded input cards
* Gradient backgrounds
* Icon-rich sidebar
* Themed buttons and layout optimization

**🔁 5. User Experience & Interactions**

Users can:

* 🔄 Switch between cities and dynamically view KPIs
* 💬 Ask follow-up questions in chat
* 🧾 Generate sustainability reports or summaries
* 🌱 Discover new eco-friendly tips
* 📊 Upload new KPI datasets for real-time forecasting

**🔌 Real-time Features Enabled By:**

* **Two-way binding** between Streamlit & FastAPI
* **Live backend JSON** integration and state updates

**✅ Ideal For:**

* Smart city governance dashboards
* Citizen engagement platforms
* Policy summarization assistants
* Educational sustainability tools